Modelling epigenetic information maintenance a Kappa tutorial

Jean Krivine, Vincent Danos, Arndt Benecke

Modelling epigenetic information maintenance a Kappa tutorial

Jean Krivine[1], Vincent Danos[2], and Arndt Benecke[1]

[1] Institut des Hautes Études Scientifiques, France
[2] University of Edinburgh

The purpose of this tutorial is to explain and illustrate an approach to the quantitative modelling of molecular interaction networks which departs from the usual notion of (bio-) chemical reaction. This tutorial is self-contained and supposes no familiarity with molecular biology.[1]

We shall use a modelling language called Kappa [1], but much of what we will present equally applies to the larger family of rule-based modelling frameworks -and in particular to the BNG language [2] which is very close to Kappa. For a technical exposition of Kappa as a stochastic graph rewriting system, the reader can consult Ref. [3].

To demonstrate the interest of a rule-based approach we will investigate a concrete biological question, that of the maintenance of epigenetic information.

Our plan is to:
- articulate in purely biological terms an epigenetic repair mechanism (§1)
- capture this mechanism into a simple rule-based model (§2)
- equip this rule set with numerical information (rule rates, copy numbers of various intervening agents) to obtain a quantitative model (§3.1-3.2)
- exploit the said model by investigating various questions (§3.3-3.4)

Although the model we present here is congruent with the current evidence, and generates interesting results, it offers a -perhaps overly- simplified view of the relevant biological mechanisms. Its primary use is to introduce gradually the Kappa concepts and (some of its) methods in a concrete modelling situation, as we make progress in our plan.

1 Epigenetic repair

Key epigenetic information is encoded in the human genome via a chemical modification of C bases into their methylated form mC. The resulting methylation patterns which are far from random, can be thought of as annotations of the DNA which determine the shutdown of their associated DNA segments. Such bookmarkings are inherited upon duplication -hence the qualifier epigenetic. This large-scale mechanism to manage the genome plays an important role in cell differentiation and unsurprisingly is very often found disrupted in cancers. The mCs form about 2% of the total number of Cs and are recognised by a suitable machinery -among which the MeCP2 protein- which drives the DNA

[1] The reference model can be obtained at `krivine@ihes.fr`, and the Kappa implementation used for this paper at `support@plectix.com`.

compaction and subsequent gene silencing. Both the setting and maintenance of epigenetic information is under intense investigation [4]. In this note we will concentrate on the maintenance aspects.

Indeed, maintenance or repair is needed since there are problems inherent to the low-level biochemical substrate of epigenetic information. DNA base pairs can be either of the AT type or the CG one (Fig. 1), and as said, the latter can be further modified as mCG. The problem is that mCGs and CGs can endure spontaneous chemical transitions to TG and UG mismatches (roughly four times per second per genome). If not reset to their respective original values, such mismatches would inevitably lead to erratic and eventually damaging genetic expression profiles. It must be that there are various agents at work within the cell in charge of recognising and repairing these mismatches and thus stabilizing the epigenetic information.

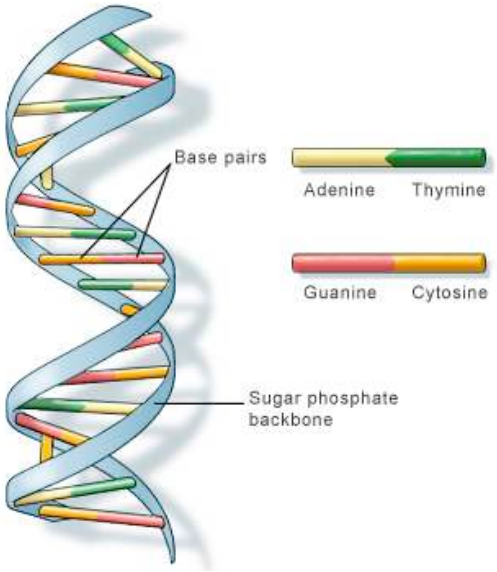

Fig. 1. *The dual base pairs AT and CG provide a redundant representation of genetic information; not so for epigenetic information which is not redundantly represented in its sole biological mC substrate (Image from the U.S. National Library of Medicine).*

1.1 The repair problem

Recent findings point at a surprising fact, namely that both kinds of mismatches seem to be recognised by the *same* agent TDG which is also in charge of excising the faulty bases T and U, before they can be replaced. Considering that after

excision there is no way to tell from the local DNA state itself what the nature of the mismatch was (in sharp constrast with the redundancy of the representation of genetic information as in Fig. 1), one wonders how a proper resetting can ensue.

Let us examine the life cycle of C bases in more details to see what the problem is. As said, Cs and mCs are subject to spontaneous deaminations into U and T respectively. These are happening roughly at a rate of 1 per second and per billion bases (which give means of calibrating the time units used in the model, of which more in §3). As shown Fig. 2, the enzymatic repertoire of the host cell shows no way how to directly reverse those changes. Instead, the C life cycle has this enigmatic feature that both U and T converge to C after being processed by TDG and APE1. In other words the cycles used to reset faulty bases at their initial methylation state join ambiguously at a base pair CG. There the system in a state where it is unclear whether the last step -performed by Dnmt3A- should be taken, ie whether C should be methylated.

The system has no obvious local memory and is in the danger of making two kinds of mistakes, either by methylating a C that was not, or by forgetting to re-methylate a C that was.[2]

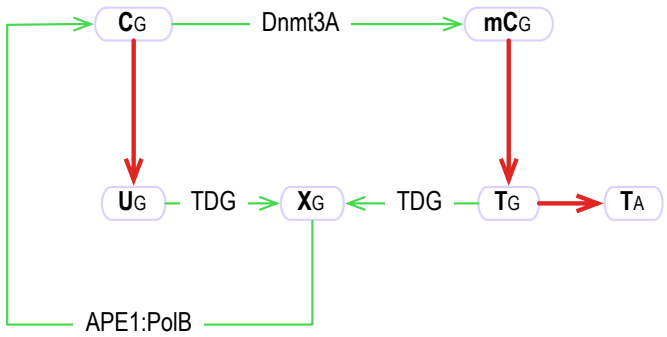

Fig. 2. *C's deamination and repair cycle: Cs and methylated Cs are subject to deaminations into U and T; TDG can recognise and excise the induced mismatches (X stands for the lack of a base), while APE1:PolB can put back in place a C, and Dnmt3A can transfer the methyl group that is needed to make it an mC. The question is how does Dnmt3A know whether a C should be remethylated or left as is?*

1.2 A solution?

From the above it clearly follows that some memory of the mismatch type must be established before the excision of the faulty base by TDG. DNA-processing

[2] In fact there is a third possible mistake which is for the BER complex (base excision repair machinery) to correct the TG mismatch into a TA, but we are not taking this into account.

proteins are not monolithic objects and can be usefully decomposed into sub-units providing different functionalities and usually called domains.

The structure of TDG reveals two DNA binding domains:
- *N140* (named after its position in TDG's amino-acid chain) responsible for the DNA-binding and excision activity,
- and *rd* (regulatory domain) which provides a second means to bind DNA.

So it seems quite natural to hypothesize that:
- TDG binds both mismatches using its *N140* DNA-binding domain,
- TDG uses another domain *rd* to bind the DNA a second time in the *specific* case of a TG mismatch.

Such a mechanism leaves the system in an unambiguous state even after the excision of the faulty bases T or U -provided that the excision of a TG mismatch is only performed after the secondary TDG binding via *rd* has happened. This we will also suppose. We will refer to this mechanism as the transient memory assumption, and refer to the *rd*.

Note that the memory of the type of mismatch under repair is kept -not in the DNA substrate itself- but rather in the transient assembly of some of the repair agents. In other words the redundancy is buily dynamically during by the repair process itself. This is compatible with the now emerging picture of DNA-processing complexes as being assembled only ever partially, and with causal constraints implemented via enzymatic steps [5]. Another thing the reader might already have noticed, is that an immediate consequence of our hypothetic mechanism is that a knock-out of the *rd* domain on TDG should hinder the repair of mCs, and potentially lead to their complete loss. This is indeed observed experimentally. What is also known is that the TG mismatch is a much stronger perturbation of the DNA structure than the UG one, and hence it is plausible to suppose as we do here that TDG can tell the two apart.

In order to provide further support one could try out various other experiments as in the course of a normal biological investigation. However, at this stage it might be perhaps wiser and more economic to provide a quantitative model prior to any further experimental elaboration. Indeed, it is easy to get carried away and convince oneself of the general good-lookingness of an informal hypothesis. Not so with a numerical model which is a stronger stress test as it incorporates a proof of quantitative consistency of our starting assumption (of course not by any means a proof that the said assumption is indeed true). Constructing such a model is what we do in the next two sections (§2–3).

2 The qualitative model

One first difficulty in turning our informal assumption above into quantitative form is not even quantitative -as it is a matter of pure representation. The various agents involved in epigenetic repair will associate in so many different contexts that requiring a model to explicit all of these is unrealistic. A quick glance at Fig. 3 describing the domains and interactions of CBP, one of the biological agents we will be concerned with, reveals the potential combinatorics involved.

Such cases are in no way exceptional. In fact, this combinatorial complexity is often amplified by the fact that proteins often have multiple modification states. One would like to specify molecular events which are conditioned only on partial contextual information, that is to say one would like to use *rules* and not simple reactions.

As our mechanism is formulated directly in terms of domain-domain binding, it would also be convenient if one were to use a quantitative formalism that offers binding as a primitive operation. The language Kappa -a stochastic calculus of binding and modification introduced in the context of cellular signalling networks- fits such representational needs well, and we shall use it here.

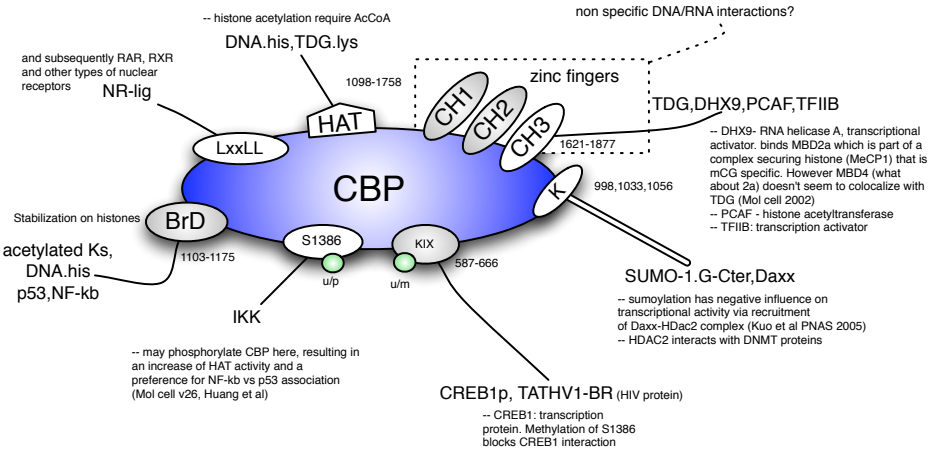

Fig. 3. *Domains and interactions of CBP: the fine domain structure of CBP is far richer than what we will represent in our model (§2.3).*

2.1 Agents

To begin with, we have to list the various agents that we want to describe in our model and decide at which level of resolution each will be made available.

Folowing Fig. 2, we will thus use the following inventory of agent types (of which there will be many copies in a given state of the system):
- an agent DNA representing a unit of closure and opening (one can think of it as a DNA double strand about a kilobase long)
- a pair of agents MeCP2, CBP controlling DNA segments closure and opening
- an agent TDG in charge of recognising/excising both types of mismatches
- a combined agent APE1:PolB to fill in the lacking C after excision
- and Dnmt3A to methylate Cs

Each of the agents above is equipped with an interface, that is to say a set of sites. Sites are a generic abstraction for meaningful subunits of an agent such

a chemically modifiable site, a binding site, etc. As shown in Fig. 4 where all six agents are represented with their sites, some of these sites can bind together (note that the curvature of the various edges carries no signification and is purely there for aesthetic reasons). The resulting site graph is called the model *contact map*. As we wish to build a simple model, not all known sites are included in our map. For instance, CBP has several binding sites (Fig. 3) and yet in our map we consider only one for binding compact chromatin and one for binding TDG.

A *state* of our model is any site graph, where each site is bound at most once and in a way that is compatible with the contact map, and each site has a definite internal state (if it has internal states at all). It is important to realise that the contact map does not specify under which conditions a binding/unbinding or a modification is possible, it merely registers which bindings are possible at all.[3] It is the role of rules to specify for each possible binding, unbinding or internal state modification, under which partial conditions it happens. We will describe rules in the next subsection.

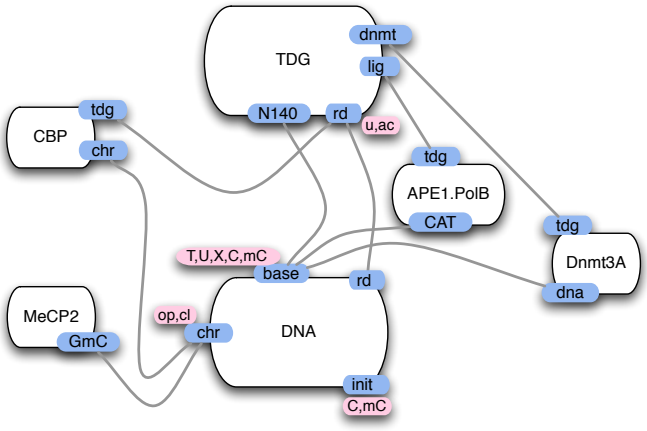

Fig. 4. *The model's contact map: agents figure together with their sites and the potential values held by these sites (only TDG and DNA have sites holding values). Potential bindings are represented by edges.*

The DNA *init* site in the contact map has no biological meaning and is used to keep a record of the base initial value and track repair mistakes (see §3 below). The DNA site *chr* abstracts the complex closure and opening mechanism of DNA segment; *chr* stands for chromatine which is the biological name for the DNA molecule and its associated cortege of structure-managing proteins, it can be either closed (compact) or open.

[3] Genome-wide contact maps are beginning to appear improving spectacularly the level at which the mass action proteic systems can be described [6].

2.2 Rules

The language of rules on (site) graphs that we will be using to describe the dynamics of our system of interest can be neatly described in mathematical form, but we will not belabour this point in this tutorial and keep an intuitive approach.

We shall only consider here the essential rules, ie the ones that are directly in charge of setting and exploiting the transient memory. There are several other rules in the full model including the spontaneous deamination rules, as well the rules associated to the chromatin control, the APE1:PolB base synthesis rule, and those controlling the association of TDG with its various partners other than DNA, but these pose no particular problem and are not shown.

What we need first is a pair of *recognition rules* stipulating how TDG recognises the DNA mismatches, and how TDG tells apart the two kinds of mismatches. These two rules are represented Fig. 5 and embody half of our transient memory assumption. This is of course a very simplified view as in reality, it might well be that TDG can bind open DNA unspecifically and diffuse along the DNA strand [4]. Subtler and more realistic behaviours of this sort could be incorporated in the model, and one would have to concatenate explicitely our DNA segments, and specify rules for sliding. As said, for this tutorial we shall keep things simple.

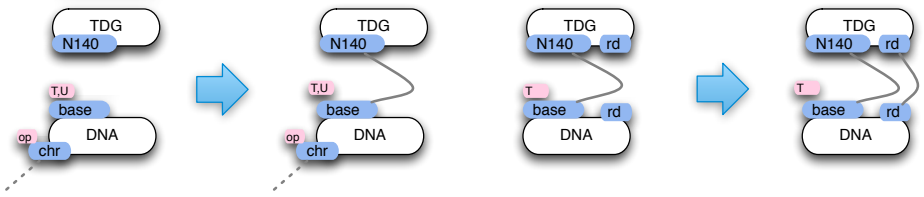

Fig. 5. *TDG-DNA recognition rules: the dotted semi-edge means that the binding state of DNA(chr) is left unspecified, it can be free or not; the key second binding depends on the base being T (hence having been mC); the first binding presupposes that the chromatin is opened (the chr site holds the value 'op').*

Then we need a pair of *excision rules* where TDG bites off the faulty base and brings along APE1:PolB to place a C at the place left vacant. These two rules are given Fig. 6. We could have decomposed these rules by separating the excision step from the binding exchange one. It is worth noticing that the excision rule in the T case does require the binding on the *rd* domain to be in place, as excising to soon would compromise the arming of our temporary memory, that is to say the *rd*-mediated binding to DNA. This is the other half our assumption.

To complete the repair triptych we need a *remethylation rule* where Dnmt3A comes into play. The rule is given Fig. 7, and as we can see it conditions this event on the presence of the memory binding at *rd*. It is interesting to see that

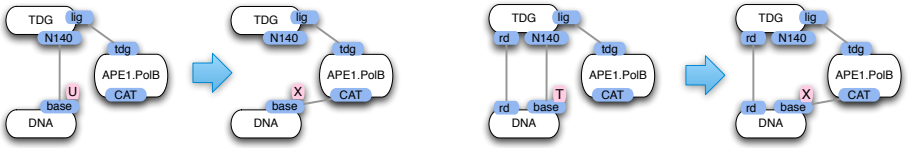

Fig. 6. *TDG:DNA-APE1:PolB excision rules (abbreviated respectively as "repair U" and "repair T" below): TDG and APE1:PolB exchange connexions to the base under repair in order for APE1.PolB to replace the missing base, that was excised by TDG; importantly, and in accordance our basic assumption (§2), the excision of a T (second rule) presupposes the binding of rd.*

at this stage we meet with the problem dual to the one dealt with the T excision rule Fig. 6. Namely, we would like to make sure that the memory binding does not disappear too soon.

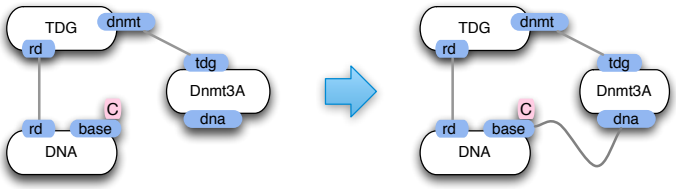

Fig. 7. *DNA-Dnmt3A remethylation rule: Dnmt3A binds to the base left free by TDG and remethylates it; the memory is not local to any of the agents but resides in their assembly.*

A way to do this is to let go of the *rd* binding only after remethylation of the base. This is easy to formulate as a rule but we choose not to because it seems too unrealistic. For one thing the *rd* binding is known to be weak experimentally, and it seems unlikely that the excision and subsequent loss of the *N140* binding will help stabilize it. Even if that were the case, it seems that asking TDG to know if the base it is *no longer bound to* is methylated is too non-local an interaction. The good news is that this is actually not necessary and one can let this binding be quite weak without compromising the performance of repair.

We can only (and will in §3) elaborate on this point when we have a quantitative model. Indeed, rules in the absence of any kinetic information only specify a non-deterministic system. If we suppose as we do that the *rd* binding is even somewhat reversible, nothing prevents our memory binding to dissolve. The non-deterministic transition system associated to our rule set is wrong, it will make mistakes. Numerically however, such mistakes just never happens and the repair system is correct. This discussion begs the question of how one equips the model quantitatively, a matter to which we turn in the next section.

2.3 Stories

Before turning to the quantitative aspect, however, we can check the causal soundness of the rule set we have put together. We ask what minimal trajectories starting from a mC deamination event will lead to its proper repair. As in any rewriting system, one has a natural notion of commuting/concurrent events, which one can use to simplify trajectories leading to event of a given type - here a C remethylation- by eliminating spurious concurrent events. In practice this causal simplification leads to an overwhelming number of thumbnails. But one can simplify them further by asking that they contain no subconfiguration leading to the same observable. This notion of incompressible subtrace, where all steps matter, and which we call story, gives strong insights in the causal mechanisms of a rule set, and is a powerful tool to debug complex rule sets.

An example is given Fig. 8. The right part depicts an mC deamination with the subsequent chromatin opening by CBP, while the left part shows a Dnmt3A:TDG:APE1.PolB trimer recognising and processing the ensuing mismatch according to the rules given previously. We will see later two other variant stories (§3.4). All observed stories are suffixes of these three archetypical ones in this case.

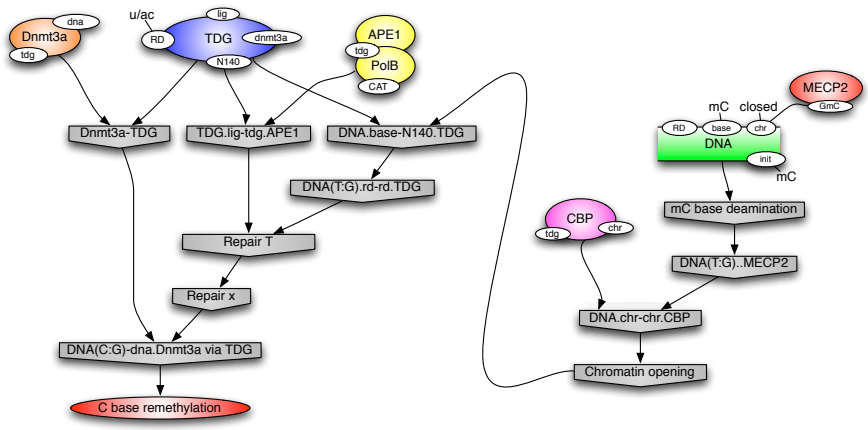

Fig. 8. *A story leading to a C remethylation: nodes represent events, that is to say rule applications; causal precedence between events is indicated with arrows, eg the chromatin opening is a necessary step for TDG to bind at N140.*

3 The quantitative model

Our rule set is a good start, however, as we just mentioned, there are pending questions that need numerical information to begin to be answered. We first need

to understand how one can use a rule set to create a dynamical system, specifically an implicit continuous time Markov chain (CTMC). By implicit, we mean that there is no need to construct the explicit state space of the system, which is fortunate since combinatorial complexity forbids an explicit representation of all but the simplest systems [7].

3.1 The CTMC

The needed CTMC structure is as follows. Define the activity of a rule r in a given state x as $\mathbf{a}(r,x) := k_r[s_r,x]$ where $k_r \in \mathbb{R}^+$ is the rule intrisic rate (a parameter), s_r is the rule left hand side, and $[s_r,x]$ is the number of matches of s_r in x. Define the global activity as $\mathbf{a}(x) := \sum_r \mathbf{a}(r,x)$. The probability that the next rule to be applied is r is given by $\mathbf{a}(r,x)/\sum_r \mathbf{a}(r,x)$, and the random time elapsed δt is given by $p(\delta t > T) = \exp(-\mathbf{a}(x)T)$.

Observe that the probability to pick r is 0 iff r has no matches, which seems logical, and likewise the expected time for anything to happen is ∞ iff $\mathbf{a}(x) = 0$.

The above dynamics implements a stochastic version of the mass action law and is often referred to as the 'Gillespie algorithm'. The behaviour of the system will depend both on the reaction rates and the copy numbers, meaning the number of agents of each type defining the system initial state. In our special case these copy numbers will be invariant since we have introduced no rules that consumes or produces an agent. Although we don't have serious quantitative data with which one could constrain our parameters, we can nevertheless make reasonable guesses.

3.2 Choosing parameters

Let us start with copy numbers. We specify them by annotating the contact map as in Fig. 9. Furthermore, we suppose that all agents are disconnected in the initial state, except for the 400 closed DNA agents which we suppose dimerized with a MeCP2. This is just a convenience since what interests us is the behaviour of the system at steady state and the particulars of the initial state will soon be forgotten -except for the copy numbers which as said are invariant. The 50/1 ratio of Cs to mCs is respected. The other copy numbers are chosen so that the total number of repair agents is about 1% of the number of DNA segments. The true experimental numbers are not known but the proportions should be about right. At any rate, with such a choice repair does not become too easy, as it would if we had more repair agents.

Let us fix (arbitrarily) the deamination rate to $10^{-2}s^{-1}$. This amounts to defining the time units of the model. Since we have roughly $20,000$ DNA agents, one will see about 200 deaminations per time unit, which is 50 times more than in the genome, hence our time currency is worth 50 seconds, and simulations running for 500 such time units (as below §3.3) should make zero mistakes.

Regarding the choice of association rates, also known as *on-rates*, eg the rate of the first TDG recognition rule given earlier, we can say the following. In general on-rates are only dependent on the diffusivity of the agents taking part in

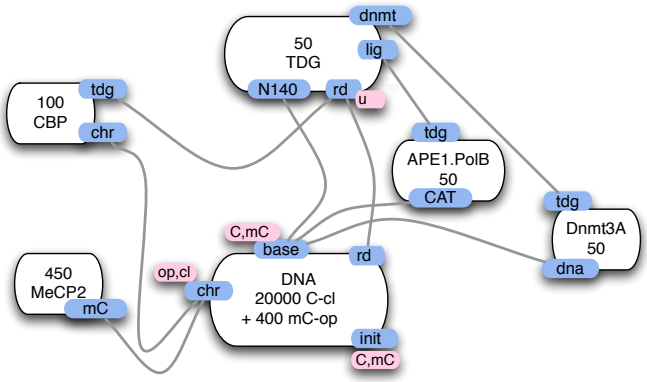

Fig. 9. *The initial copy numbers: we suppose all 400 DNA closed agents have an mC base and are dimerized with a MeCP2, the 20000 remaining DNA agents are opened and with a C base; apart from the closed DNAs all agents are disconnected.*

the interaction, so it seems reasonable to choose them all equal, in the first instance, and we will choose them all equal to 1 in our case. However, something interesting happens here. Since we are working with rules which generate a contact map that allows cyclic bindings, it is possible that in particular matches, or instances, of a given association rule, the association occurs between agents that are already connected with one another. Fig. 10 gives an example. The top most rule is the basic rule for the TDG:APE1.PolB association and various possible instances are given below. In the bottom-most one, our two agents are already connected. In such cases, called sometimes unimolecular or unary instances, obviously the rate is no longer defined by diffusion, and we shall choose these unary on-rates in the $[10^3, 10^5]$ range.

Now what of the off-rates? Contrary to on-rates, these might be much more specific to the actual binding. In the present model we have set all off-rates to 10^{-3}, except for that of the *rd*:*rd* binding which is known to be a weak binding and set to 1, and that of the *N140*:*base* binding which is known to be a strong binding and set to 10^{-5}.

Finally there is one key place in the model, namely the Dnmt3A:TDG dissociation, where we modulate the off-rate depending on the context. The reason is the following. The number of mCs being much smaller than the number of Cs, the Dnmt3As will mostly reside with TDGs that are repairing a C, ie where they are not needed. Even if we were to make TDG wait for Dnmt3A on the mCs -which we are not as explained earlier (§2.2)- this would delay repair and make it less efficient. The trick is to decrease (increase) the dissociation rate of Dnmt3A when bound where it is (not) needed, ie on an *rd*-bound (-free) TDG. This kind of honey pot technique based on rule refinement is analysed in details in Ref. [8]. If Dnmt3A saturates TDG, as is the case with the initial state defined above, it does not make a difference; however, if we suppose we have only a 1/5 ratio of Dnmt3A/TDG and do not use this trick the model performance collapses.

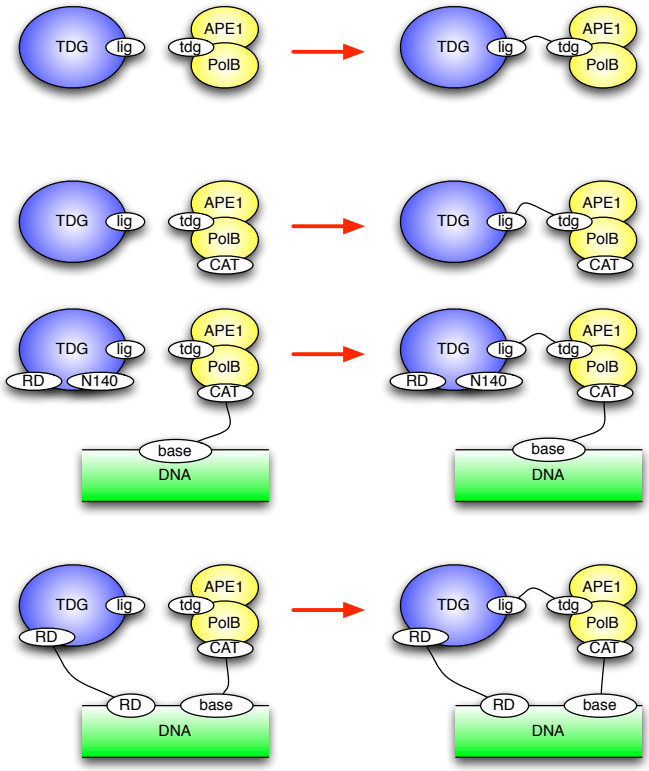

Fig. 10. *Refinement of the TDG:APE1.PolB association rule: the top rule is the basic one, the bottom three offer refined and mutually exclusive variants of the former. In the last one the two agents are already connected which makes a unary/unimolecular instance. Given the invariants of the rule set and initial state, these actually cover all possible cases of application of the top rule.*

3.3 Results

With our model numerically equipped we can now test its performances. For a repair mechanism there are two clear observables of interest, its accuracy and efficiency. The former measures how often a mistake is made, while the latter measures how long the repair queue is. Let us see if our transient memory model finds the correct trade-off here, as a substantial part of the numerical proof of concept we are looking for consists precisely.

As we can see on Fig. 11, the model does one mistake for the entire duration of this particular stochastic simulation. In general one onbserves less than one mistake. On the other hand the size of the mC repair queue stabilizes at about 50% of the mC population. Regarding the repair of Cs, the repair queue is kept well below 1% of the population (not shown). What is remarkable because it is

counter-inuitive is that the *rd* affinity is three orders of magnitude weaker than others in the model.

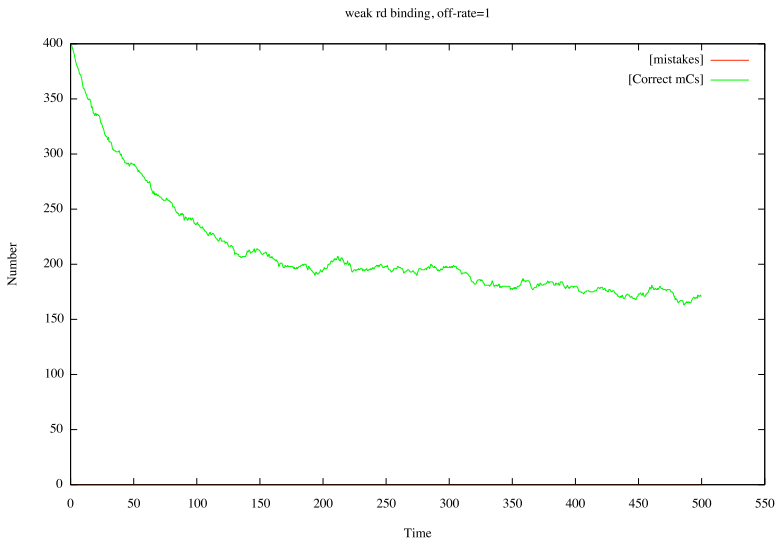

Fig. 11. *The curve represents the number of repaired mCs (the remainder are being repaired); one observes no mistakes -defined as DNA agents with a C base that was methylated (which we can know by looking at the agent init site) with a free rd site.*

To verify that we have struck the correct trade-off, we can modify the *rd* off-rate. If as intuition would have it, we decrease it, that is to say we increase the affinity of the *rd* binding, then the accuracy does not suffer, but the efficiency is considerably lowered as one can see on Fig. 12 (and the same happens for the repaired *C*s, not shown). If one the other hand one lowers the affinity even more, then mistakes start to accumulate as one sees in Fig. 13.

3.4 Stories (continued)

We have shown earlier how stories can serve as a useful window into the causal structure of a model. With a numerical model we can put then to further use, by asking for their frequence. For the present model, with its reference parameters, we find that we can classify stories as suffixes of three archetypical stories. One which we have already seen in Fig. 8 occurs about 90% of the time. The other two are shown Fig. 14 which occurs about 9%, and Fig. 15 about 1%. Observe that in both variants, the TDG agent that is responsible for the repair has first to dissociate from another DNA agent before coming to the rescue of the one in point. This hints at a tension in the repair system where the supply in TDG is lower than the demand -a tension of which we have seen the consequences on

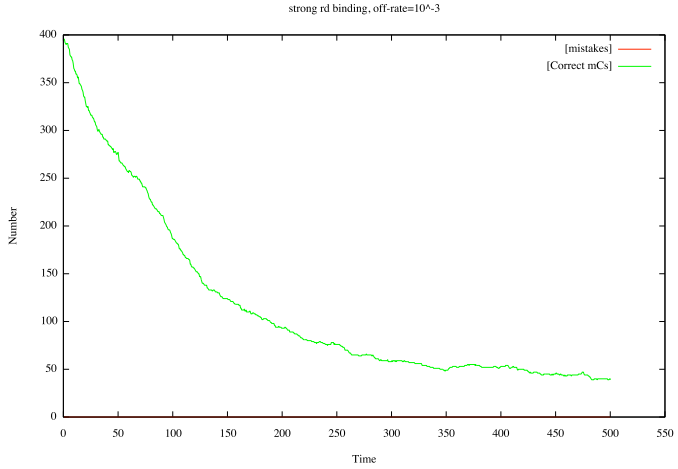

Fig. 12. *The refence model perturbed by increasing the rd affinity: still no mistakes but the number of repaired mCs decreases dramatically.*

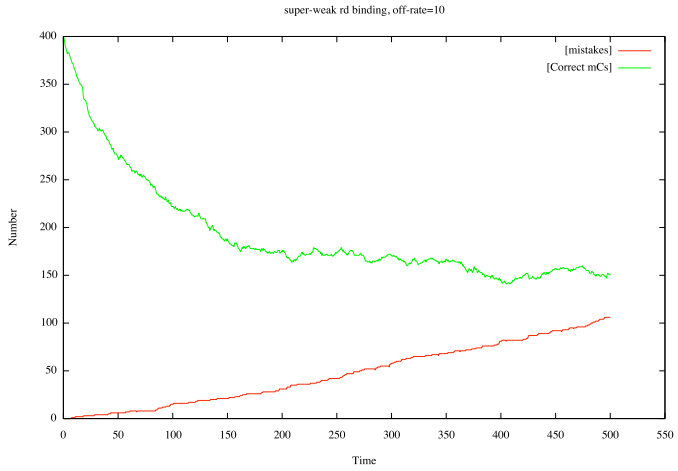

Fig. 13. *The refence model perturbed by decreasing the rd affinity even further: the number of mCs under repair hardly changes in our time window, but the number of mistakes increases steadily.*

the optimal affinity of the *rd* domain. In this kind of modelling context, where the just-in-time assembly of complexes is key to the operation of the system, being able to peek in the individual agent trajectories is not only a good way to understand how the rules in a rule set combine, but also offers a way to understand what are the key parameters which will impinge the most on the system behaviour.

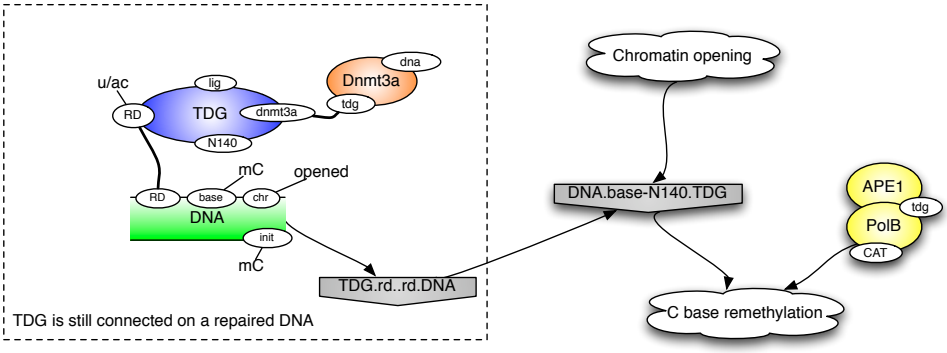

Fig. 14. *Another (less frequent) story leading to a C remethylation, only the sections that differ from the preceding one (Fig. 8) are shown. One sees that TDG is busy at the outset repairing an mC.*

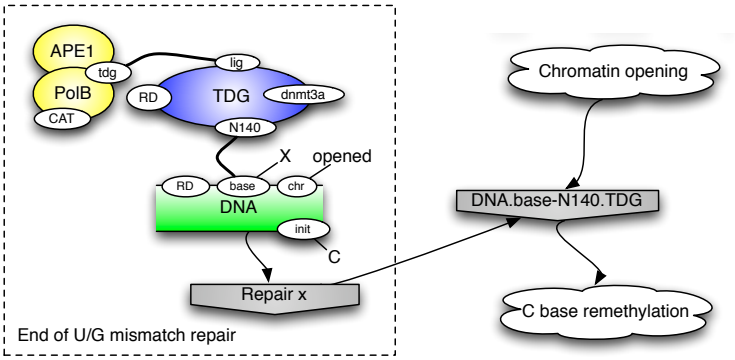

Fig. 15. *Another (even less frequent) story leading to the same conclusion where TDG is busy repairing a C.*

4 Conclusion

To be clear, the model we have considered fails to be realistic in many ways, and it is only one possible model of the system of interest. We have tried to keep rules within the plausible (eg see the discussion at the end of §2.2) but for instance the affinities between TDG and its helper agents APE1:PolB and Dnmt3A is implausibly strong at the moment. Likewise the numbers of copies of agents has little experimental basis. The model also fails to incorporate a lot of the complexities that we know of regarding the biology of TDG and in particular the role of acetylation and sumoylation which is yet to be understood. Nevertheless the somewhat paradoxical conclusion that the *rd* affinity has to be quite a bit weaker than others is confirmed by experimental data.

This gives some degree of confidence in the basic assumption which one can think of extending in many ways to put other more sophisticated hypotheses to

test. For instance, one should consider the resilience of the model to bursts of UG mismatches as are likely to be revealed by the opening of chromatin subsequent to some mC being deaminated. Such repair shocks because of the 50/1 ratio of Cs to mCs might be difficult to cope with. Other extensions worth pursuing are the competition with the BER (base excision repair) machinery and the potential drift to TA mistakes (as explained briefly in an earlier footnote, §1.1), and/or the interaction with transcriptional mechanisms which might shed some light on transcriptional leakages whereby one sees genes expressed that should presumably be shut in compact chromatin. Indeed the queuing of mCs under repair might allow the opportunistic transcription of hidden genes.

Further, and beyond the particulars of the present biological situation, it is reassuring to see that using the proper approach, it is actually possible to make way in the modelling of systems where binding figures prominently and combinatorially -and which are not well-understood yet. Kappa gives the means and in some sense the imagination to represent and capture numerically assumptions that are very natural (as our transient memory assumption) and difficult to handle otherwise. This suggests that a modelling activity could be succesfully pursued at the same time and in the same stride as experiments.

References

1. Jerome Feret, Vincent Danos, Russell Harmer, Jean Krivine, and Walter Fontana. Internal coarse-graining of molecular systems. *PNAS*, Apr 2009.
2. James R. Faeder, Michael L. Blinov, and William S. Hlavacek. *Systems Biology*, volume 500, chapter Rule-based modeling of biochemical systems with BioNetGen, pages 113–167. Humana Press, 2009.
3. Vincent Danos, Jerome Feret, Walter Fontana, Russell Harmer, and Jean Krivine. Rule-based modelling, symmetries, refinements. In Springer, editor, *FMSB 2008*, volume 5054 of *LNBI*, pages 103–122, Jun 2008.
4. Manel Esteller. Cancer epigenomics: Dna methylomes and histone-modification maps. *Nat Rev Genet*, 8(4):286–298, Apr 2007.
5. Christoffel Dinant, Martijn S Luijsterburg, Thomas Höfer, Gesa von Bornstaedt, Wim Vermeulen, Adriaan B Houtsmuller, and Roel van Driel. Assembly of multiprotein complexes that control genome function. *J Cell Biol*, 185(1):21–6, Apr 2009.
6. Philip M Kim, Long J Lu, Yu Xia, and Mark B Gerstein. Relating three-dimensional structures to protein networks provides evolutionary insights. *Science*, 314(5807):1938–1941, 2006 Dec 22.
7. Vincent Danos, Jérôme Feret, Walter Fontana, and Jean Krivine. Scalable simulation of cellular signaling networks. In Z. Shao, editor, *Proceedings of APLAS 2007*, volume 4807 of *LNCS*, pages 139–157, Nov 2007.
8. Vincent Danos, Jerome Feret, Walter Fontana, Russell Harmer, and Jean Krivine. Investigation of a biological repair scheme. In G. Paun, editor, *Proceedings of WMC'09*, volume 5391 of *LNCS*. Springer, Jan 2009.